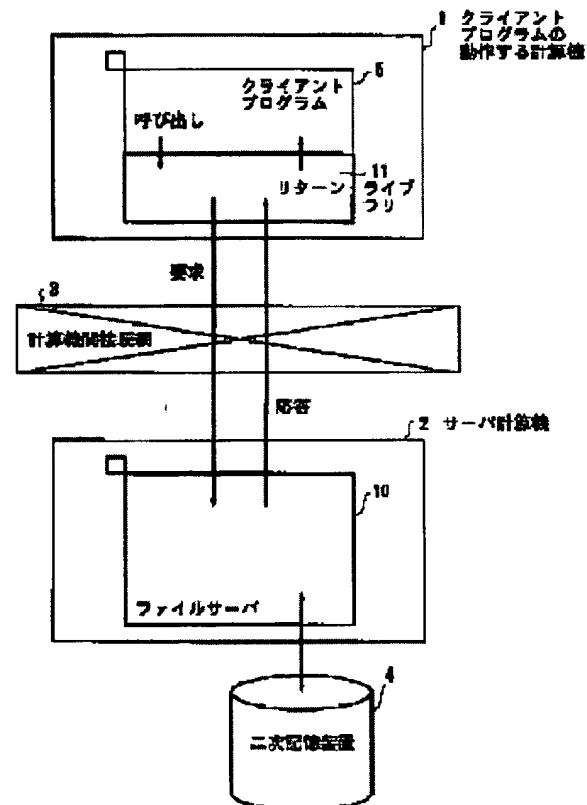


REMOTE ACCESS METHOD FOR SECONDARY STORAGE DEVICE**Publication number:** JP9198336**Publication date:** 1997-07-31**Inventor:** AIBA YUICHI**Applicant:** NIPPON ELECTRIC CO**Classification:** 2**- International:** G06F12/00; G06F13/00; G06F12/00; G06F13/00;
(IPC1-7): G06F13/00; G06F12/00**- european:****Application number:** JP19960027263 19960122**Priority number(s):** JP19960027263 19960122

Report a data error here

Abstract of JP9198336

PROBLEM TO BE SOLVED: To shorten the read/write processing time to a remote secondary storage device through an inter-computer connection network. **SOLUTION:** A library 11 linked to a client program 5 divides a read/write call from a client program to the remote secondary storage device 4 into plural read/write requests and transmits the whole to a server computer 2 at the same time through the inner-computer connection network 3. The file server 10 of a server computer 2 multiplexes and executes a processing of receiving the read/write requests, accessing to data on the requested secondary storage device 4 for one of the received read/write requests and transmitting one response to the client program 5 for a request source for the plural read/write requests. Thus, the communication time between the computers is overlapped with physical read/write processing time to the secondary storage device 4 so as to shorten read/write processing time to the remote secondary storage device.



Data supplied from the esp@cenet database - Worldwide

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平9-198336

(43) 公開日 平成9年(1997)7月31日

(51) Int.Cl. ⁸	識別記号	片内整理番号	F I	技術表示箇所
G 0 6 F 13/00	3 5 7		G 0 6 F 13/00	3 5 7 Z
12/00	5 4 5		12/00	5 4 5 A

審査請求 有 請求項の数 5 F D (全 21 頁)

(21) 出願番号 特願平8-27263

(22) 出願日 平成8年(1996)1月22日

(71) 出願人 000004237

日本電気株式会社

東京都港区芝五丁目7番1号

(72) 発明者 相場 雄一

東京都港区芝五丁目7番1号 日本電気株式会社内

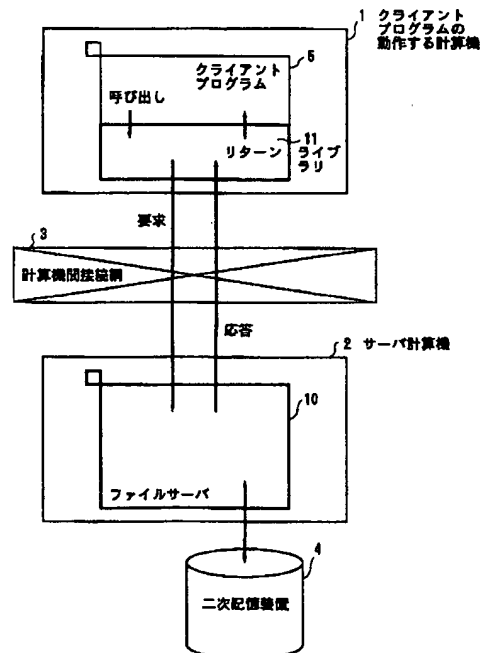
(74) 代理人 弁理士 境 廣巳

(54) 【発明の名称】 二次記憶装置への遠隔アクセス方法

(57) 【要約】

【課題】 計算機間接続網を介した遠隔二次記憶装置へのリード／ライトの処理時間を短縮する。

【解決手段】 クライアントプログラム5からの遠隔二次記憶装置4へのリード／ライト呼び出しに対し、クライアントプログラム5にリンクされたライブラリ11は、それを複数のリード／ライト要求に分割して計算機間接続網3を介してサーバ計算機2に一度に全部送信する。サーバ計算機2のファイルサーバ10は、リード／ライト要求を受け取り、受け取ったリード／ライト要求1つに対し、要求された二次記憶装置4上のデータにアクセスをした後、要求元のクライアントプログラムに回答1つを送信するという処理を、複数のリード／ライト要求に対し、多重化して実行することにより、計算機間の通信時間が二次記憶装置4への物理リード／ライト処理時間に重なるようにして、遠隔二次記憶装置へのリード／ライトの処理時間を短縮する。



【特許請求の範囲】

【請求項1】 第1の計算機上で動作するクライアントプログラムから、前記第1の計算機と計算機間接続網を通じて相互に通信可能な第2の計算機に接続されている二次記憶装置上のデータに対してアクセスする方法であって、

前記第1の計算機においては、

前記クライアントプログラムによるリード／ライト呼び出しを、一定サイズのデータへのリード／ライト要求に分割し、該分割して得られたリード／ライト要求を全部一度に前記第2の計算機に送信し、

前記送信した全てのリード／ライト要求について前記第2の計算機から応答が返却されていることを確認した後、前記クライアントプログラムにリード／ライト呼び出しの結果を戻し、

前記第2の計算機においては、

前記第1の計算機からのリード／ライト要求を受け取り、受け取ったリード／ライト要求1つに対し、要求された二次記憶装置上のデータにアクセスをした後、要求元のクライアントプログラムに応答1つを送信するという処理を、複数のリード／ライト要求に対し、多重化して実行することを特徴とする二次記憶装置への遠隔アクセス方法。

【請求項2】 第1の計算機上で動作するクライアントプログラムから、前記第1の計算機と計算機間網を通じて相互に通信可能な第2の計算機に接続されている二次記憶装置上のデータに対してアクセスする方法であって、

前記第1の計算機においては、

前記クライアントプログラムによるリード／ライト呼び出しを、一定サイズのデータへのリード／ライト要求に分割し、

該分割して得られたリード／ライト要求の数が予め定められたスレッシュールド値以下のときは全部一度に前記第2の計算機に送信し、スレッシュールド値を超えているときは最初にスレッシュールド値分のリード／ライト要求を前記第2の計算機に送信した後、リード／ライト要求に対応する前記第2の計算機からの応答を1つ確認する毎に残りのリード／ライト要求を1つ前記第2の計算機に送信し、

前記送信した全てのリード／ライト要求について前記第2の計算機から応答が返却されていることを確認した後、前記クライアントプログラムにリード／ライト呼び出しの結果を戻し、

前記第2の計算機においては、

前記第1の計算機からのリード／ライト要求を受け取り、受け取ったリード／ライト要求1つに対し、要求された二次記憶装置上のデータにアクセスをした後、要求元のクライアントプログラムに応答1つを送信するという処理を、複数のリード／ライト要求に対し、多重化し

て実行することを特徴とする二次記憶装置への遠隔アクセス方法。

【請求項3】 第1の計算機上で動作するクライアントプログラムから、前記第1の計算機と計算機間網を通じて相互に通信可能な第2の計算機に接続されている二次記憶装置上のデータに対してアクセスする方法であって、

前記第1の計算機においては、

前記クライアントプログラムによるリード／ライト呼び出しを、一定サイズのデータへのリード／ライト要求に分割し、

該分割して得られたリード／ライト要求の送信に先立って前記第2の計算機に対して予約の要求を送信し、前記送信した予約の要求に対する前記第2の計算機からの応答に含まれる予約数からスレッシュールド値を決定し、

前記分割して得られたリード／ライト要求の数が前記スレッシュールド値以下のときは全部一度に前記第2の計算機に送信し、スレッシュールド値を超えているときは最初にスレッシュールド値分のリード／ライト要求を前記第2の計算機に送信した後、リード／ライト要求に対応する前記第2の計算機からの応答を1つ確認する毎に残りのリード／ライト要求を1つ前記第2の計算機に送信し、前記送信した全てのリード／ライト要求について前記第2の計算機から応答が返却されていることを確認した後、前記第2の計算機に対して予約解除を要求すると共に、前記クライアントプログラムにリード／ライト呼び出しの結果を戻し、

前記第2の計算機においては、

前記第1の計算機からの予約の要求に対し、自計算機の現在の負荷状況に応じて1つのクライアントプログラムに対して許容するリード／ライト要求数を予約数として前記第1の計算機に通知すると共にその予約数に見合ったメモリ量を確保し、

前記第1の計算機からのリード／ライト要求を受け取り、受け取ったリード／ライト要求1つに対し、要求された二次記憶装置上のデータにアクセスをした後、要求元のクライアントプログラムに応答1つを送信するという処理を、複数のリード／ライト要求に対し、多重化して実行し、

前記第1の計算機からの予約解除の要求に対し、前記確保していたメモリ量を解放することを特徴とする二次記憶装置への遠隔アクセス方法。

【請求項4】 第1の計算機上で動作するクライアントプログラムから、前記第1の計算機と計算機間網を通じて相互に通信可能な第2の計算機に接続されている二次記憶装置上のデータに対してアクセスする方法であって、

前記第1の計算機においては、

前記クライアントプログラムによるリード／ライト呼び

出しを、一定サイズのデータへのリード/ライト要求に分割し、
 該分割して得られたリード/ライト要求のうちの1つの
 リード/ライト要求に予約の要求を付加した予約+リード/ライト一括要求を前記第2の計算機に対して送信し、
 前記送信した予約+リード/ライト一括要求に対する前記第2の計算機からの予約応答に含まれる予約数からスレッシュホールド値を決定し、
 前記分割して得られたリード/ライト要求の残りの数が前記スレッシュホールド値以下のときは、1つのリード/ライト要求に予約解除の要求を付加して予約解除+リード/ライト一括要求とした上で、全部一度に前記第2の計算機に送信し、スレッシュホールド値を超えているときは最初にスレッシュホールド値分のリード/ライト要求を前記第2の計算機に送信した後、リード/ライト要求に対応する前記第2の計算機からの応答を1つ確認する毎に残りのリード/ライト要求を1つ前記第2の計算機に送信していき且つ最後の1つのリード/ライト要求は予約解除の要求を付加した予約解除+リード/ライト一括要求として送信し、
 前記送信した全ての要求について前記第2の計算機から応答が返却されていることを確認した後、前記クライアントプログラムにリード/ライト呼び出しの結果を戻し、
 前記第2の計算機においては、
 前記第1の計算機から要求を受け取り、受け取った要求が予約+リード/ライト一括要求のときは自計算機の現在の負荷状況に応じて1つのクライアントプログラムに対して許容するリード/ライト要求数を予約数として前記第1の計算機に通知すると共にその予約数に見合ったメモリ量を確保した後、要求された二次記憶装置上のデータにアクセスをして要求元のクライアントプログラムに応答1つを送信するという処理を、受け取った要求がリード/ライト要求のときは要求された二次記憶装置上のデータにアクセスをした後、要求元のクライアントプログラムに応答1つを送信するという処理を、受け取った要求が予約解除+リード/ライト一括要求のときは要求された二次記憶装置上のデータにアクセスをした後、要求元のクライアントプログラムから発生した処理中のリード/ライト要求が全部完了した時点で前記確保したメモリ量を解放するという処理を、複数の要求に対し、多重化して実行することを特徴とする二次記憶装置への遠隔アクセス方法。

【請求項5】 前記第2の計算機に、複数台の前記第1の計算機を前記計算機間接続網を通じて接続したことを特徴とする請求項1、2、3または4記載の二次記憶装置への遠隔アクセス方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は二次記憶装置への遠隔アクセス方法に関し、より詳細には、複数の計算機が計算機間接続網で接続されている環境において、或る計算機で動くアプリケーションプログラムから、別の計算機に接続されている二次記憶装置上のデータに対してアクセスする方法に関する。

【0002】

【従来の技術】別の計算機に接続されている二次記憶装置（遠隔二次記憶装置）にアクセスする方法は幾つか存在する。

【0003】1つの方法は、図17（A）に示すような、Network File System（NFS）による方法である。NFSにおけるデータアクセスは、RPCという通信プロトコルレイヤを設け、ソフトウェアで実現されている（参考文献「NFS&NIS」アスキー出版局）。

【0004】NFSでは、クライアントプログラムからのリード/ライトの呼び出しをRPC要求という形に変換し、サーバ計算機上で動作するNFSデーモンというプロセスにRPCプロトコルを使って送信する。例えば、ライト呼び出しであれば、要求と共にライトデータが送信されるが、NFSデーモンでこのデータを取り込んでから、二次記憶装置に書き込み、応答をクライアントプログラムに転送する。リード呼び出しであれば、NFSデーモンは二次記憶装置から目的とするデータを読み出し、RPC要求の応答としてデータを返却する。

【0005】クライアントプログラムからのリード/ライト呼び出しをRPC要求に変換する際は、固定サイズデータへのリード/ライト要求として複数のRPC要求に分解される。通常は、1回のRPC要求でアクセスされるデータのサイズは8KByteとなっている。アクセス要求は1つずつNFSデーモンに送信され、その応答を待ってから次のRPC要求が送信される。つまり、NFSデーモンは同時には1つのRPC要求しか処理しないため、8KByteのデータを取り込む領域さえ確保していれば良い。ただし、1つのクライアントプログラムに対して1つのNFSデーモンが占有されるため、複数のクライアントプログラムからの要求を処理するためには、サイズ計算機上にNFSデーモンを複数立ち上げておく必要がある。

【0006】別の方法として、例えば特開平4-92943号公報に見られる高速LANのディスクサーバがある。この方法では、図17（B）に示すように、計算機間接続網を高速LANとし、二次記憶装置（ディスクと限定している）を管理するディスクサーバと複数のワークステーションから構成される環境を想定している。ディスクを管理するディスクサーバは専用のハードウェアとし、複数のリード/ライト要求を同時に処理できる構造となっている。

【0007】ワークステーション上ではクライアントプ

5

ログラムが動作し、クライアントプログラムからの遠隔ディスクへのリード/ライト要求を、通信によってディスクサーバに送信する。例えば、ライト要求であれば、要求と共にデータを送信する。これに対しディスクサーバでは、受信したデータをデータバッファに一旦格納してからディスクにデータを書き込み、通信によって要求元のワークステーションに結果を通知する。また、リード要求であれば、ディスクサーバでデータバッファ中に目的のデータがあるかどうかを調べ、なければディスクからデータを読み出し、データバッファに格納する。更に、格納したデータを通信によって要求元のワークステーションに送信する。

【0008】この方法では、計算機間接続網と二次記憶装置との間でデータを中継するために、ディスクサーバ内にデータバッファを持つ。データバッファは、各ワークステーション毎に領域を区切って使用する。リード/ライト要求はフレームと言う固定長単位で行われる。

【0009】以上に挙げた2つの方法は、何れもクライアントプログラムからのリード/ライト呼び出しを、複数の固定長データへの断片的なリード/ライト要求として分解し、二次記憶装置を管理する部分（サーバ計算機、ディスクサーバ）に送信することによって、遠隔二次記憶装置へのリード/ライト呼び出しの処理を可能としている。

【0010】

【発明が解決しようとする課題】しかしながら、分解された各断片リード/ライト要求を1つずつ送信し、処理結果を待ってから次の断片リード/ライト要求を送信するため、通信を何回も行うことになる。1つのリード/ライト呼び出しの処理の流れを見ると図17(C)のようになり、（断片リード/ライト要求送信）→（二次記憶装置リード/ライト）→（応答返信）→（断片リード/ライト要求送信）→……と処理され、全ての断片リード/ライト要求が完了してからクライアントプログラムのリード/ライト呼び出しの結果が返ってくる。そのため、クライアントプログラムから見た遠隔二次記憶装置アクセスのための時間は、断片リード/ライト要求の数だけの往復の通信時間が余分にかかっているように見える。

【0011】そこで、本発明では、遠隔二次記憶装置へのデータアクセス時間における通信時間のほとんどをクライアントプログラムから見えなくし、あたかもその二次記憶装置がクライアントプログラムの動作する計算機に接続されているかのようなアクセス時間を達成することを目的とする。

【0012】

【課題を解決するための手段】本発明は上記の目的を達成するために、第1の計算機上で動作するクライアントプログラムから、前記第1の計算機と計算機間接続網を通じて相互に通信可能な第2の計算機に接続されている

6

二次記憶装置上のデータに対してアクセスする方法において、第1の計算機および第2の計算機で以下の(1)乃至(4)のうちの何れかの処理を実行するようにしている。

【0013】(1)第1の計算機は、クライアントプログラムによるリード/ライト呼び出しを、一定サイズのデータへのリード/ライト要求に分割し、該分割して得られたリード/ライト要求を全部一度に前記第2の計算機に送信し、前記送信した全てのリード/ライト要求について前記第2の計算機から応答が返却されていることを確認した後、前記クライアントプログラムにリード/ライト呼び出しの結果を戻す。他方、第2の計算機は、第1の計算機からのリード/ライト要求を受け取り、受け取ったリード/ライト要求1つに対し、要求された二次記憶装置上のデータにアクセスをした後、要求元のクライアントプログラムに応答1つを送信するという処理を、複数のリード/ライト要求に対し、多重化して実行する。

【0014】このように、クライアントプログラムによるリード/ライト呼び出しを一定サイズのデータへのリード/ライト要求に分割して全部一度に第2の計算機に送信し、第2の計算機において複数のリード/ライト要求を多重処理することにより、第2の計算機での二次記憶装置の物理リード/ライト処理と計算機間接続網での通信処理が並行して行われることになり、通信の時間のほとんどが物理リード/ライト処理の時間と重なり、全体の処理時間が短縮される。

【0015】(2)第1の計算機は、クライアントプログラムによるリード/ライト呼び出しを、一定サイズのデータへのリード/ライト要求に分割し、該分割して得られたリード/ライト要求の数が予め定められたスレッシュホールド値以下のときは全部一度に前記第2の計算機に送信し、スレッシュホールド値を超えているときは最初にスレッシュホールド値分のリード/ライト要求を前記第2の計算機に送信した後、リード/ライト要求に対応する前記第2の計算機からの応答を1つ確認する毎に残りのリード/ライト要求を1つ前記第2の計算機に送信し、前記送信した全てのリード/ライト要求について前記第2の計算機から応答が返却されていることを確認した後、前記クライアントプログラムにリード/ライト呼び出しの結果を戻す。第2の計算機は、前記(1)と同じ処理を実行する。

【0016】このように、第2の計算機で一度に受け取ることのできる要求数としてスレッシュホールド値を設定し、最初に送信するリード/ライト要求数をスレッシュホールド値以下に抑えることにより、第2の計算機での最初の負荷を一定値以下に抑えることができる。さらに、1つの応答に対して1つのリード/ライト要求を送信するので、第2の計算機で受け取っているリード/ライト要求数は常にスレッシュホールド値以下となり、引続き第2の

計算機での負荷を一定値以下に抑えることができる。

【0017】(3)第1の計算機は、クライアントプログラムによるリード/ライト呼び出しを、一定サイズのデータへのリード/ライト要求に分割し、該分割して得られたリード/ライト要求の送信に先立って前記第2の計算機に対して予約の要求を送信し、前記送信した予約の要求に対する前記第2の計算機からの応答に含まれる予約数からスレッシュOLD値を決定し、前記分割して得られたリード/ライト要求の数が前記スレッシュOLD値以下のときは全部一度に前記第2の計算機に送信し、ス
10 レッシュOLD値を超えているときは最初にスレッシュOLD値分のリード/ライト要求を前記第2の計算機に送信した後、リード/ライト要求に対応する前記第2の計算機からの応答を1つ確認する毎に残りのリード/ライト要求を1つ前記第2の計算機に送信し、前記送信した全てのリード/ライト要求について前記第2の計算機から応答が返却されていることを確認した後、前記第2の計算機に対して予約解除を要求すると共に、前記クライアントプログラムにリード/ライト呼び出しの結果を戻す。他方、第2の計算機は、第1の計算機からの予約の
20 要求に対し、自計算機の現在の負荷状況に応じて1つのクライアントプログラムに対して許容するリード/ライト要求数を予約数として前記第1の計算機に通知すると共にその予約数に見合ったメモリ量を確保し、第1の計算機からのリード/ライト要求を受け取り、受け取ったリード/ライト要求1つに対し、要求された二次記憶装置上のデータにアクセスをした後、要求元のクライアントプログラムに
30 応答1つを送信するという処理を、複数のリード/ライト要求に対し、多重化して実行し、第1の計算機からの予約解除の要求に対し、前記確保していたメモリ量を解放する。

【0018】このように、動的に変化する第2の計算機の負荷状況に応じて、1つのクライアントプログラムに対して許容する予約数を第1の計算機に知らせ、第1の計算機はその予約数をスレッシュOLD値として、以降、このスレッシュOLD値に従ってリード/ライト要求を行うことにより、第2の計算機に発生する負荷が一定値を超えないように制御することができる。従って、第2の計算機が複数のクライアントプログラムを相手にしても、第2の計算機の処理能力を超えることはない。

【0019】(4)第1の計算機は、クライアントプログラムによるリード/ライト呼び出しを、一定サイズのデータへのリード/ライト要求に分割し、該分割して得られたリード/ライト要求のうちの1つのリード/ライト要求に予約の要求を付加した予約+リード/ライト一括要求を前記第2の計算機に対して送信し、前記送信した予約+リード/ライト一括要求に対する前記第2の計算機からの予約応答に含まれる予約数からスレッシュOLD値を決定し、前記分割して得られたリード/ライト要求の残りの数が前記スレッシュOLD値以下のときは、1

つのリード/ライト要求に予約解除の要求を付加して予約解除+リード/ライト一括要求とした上で、全部一度に前記第2の計算機に送信し、スレッシュOLD値を超えているときは最初にスレッシュOLD値分のリード/ライト要求を前記第2の計算機に送信した後、リード/ライト要求に対応する前記第2の計算機からの応答を1つ確認する毎に残りのリード/ライト要求を1つ前記第2の計算機に送信していき且つ最後の1つのリード/ライト要求は予約解除の要求を付加した予約解除+リード/ライト一括要求として送信し、前記送信した全ての要求について前記第2の計算機から応答が返却されていることを確認した後、前記クライアントプログラムにリード/ライト呼び出しの結果を戻す。他方、第2の計算機は、第1の計算機から要求を受け取り、受け取った要求が予約+リード/ライト一括要求のときは自計算機の現在の負荷状況に応じて1つのクライアントプログラムに対して許容するリード/ライト要求数を予約数として前記第1の計算機に通知すると共にその予約数に見合ったメモリ量を確保した後、要求された二次記憶装置上のデータにアクセスをして要求元のクライアントプログラムに
40 応答1つを送信するという処理を、受け取った要求がリード/ライト要求のときは要求された二次記憶装置上のデータにアクセスをした後、要求元のクライアントプログラムに
50 応答1つを送信するという処理を、受け取った要求が予約解除+リード/ライト一括要求のときは要求された二次記憶装置上のデータにアクセスをした後、要求元のクライアントプログラムから発生した処理中のリード/ライト要求が全部完了した時点で前記確保したメモリ量を解放するという処理を、複数の要求に対し、多重化して実行する。

【0020】このように、予約要求とリード/ライト要求とを一括して要求する予約+リード/ライト一括要求を導入することにより、第1の計算機側では予約要求を別要求として最初に送らずに済むとともに、第2の計算機側では予約応答に引き続いてリード/ライト処理が行われるので、予約応答の通信時間を物理リード/ライトの時間と重ねることができる。また、第2の計算機での物理リード/ライトの最中に、第1の計算機から後続のリード/ライト要求の送信が行われれば、それらの通信時間も物理リード/ライトの時間に重ねることができる。さらに、予約解除要求とリード/ライト要求とを一括して要求する予約解除+リード/ライト一括要求を導入することにより、予約解除の要求を別要求として最後に送らずに済み、その分の通信時間を削減することができる。

【0021】

【発明の実施の形態】次に本発明の実施の形態の例について図面を参照して詳細に説明する。

【0022】図1を参照すると、本発明を適用した計算機システムの一例は、クライアントプログラムの動作す

る計算機1と、サーバ計算機2と、これらを相互に通信可能に接続する計算機間接続網3と、サーバ計算機2に接続された二次記憶装置4とから構成されている。

【0023】サーバ計算機2上ではファイルサーバ10と呼ぶプロセスが動作している。ファイルサーバ10は、リード/ライト要求を、通信によって受け付け、要求内容に従ってサーバ計算機2に接続されている二次記憶装置4上のデータをリード/ライトし、通信によって応答を返却する。図1の例では、汎用的な計算機上にファイルサーバ10というプロセスを動作させることによってサーバ計算機の持つべき機能を実現しているが、他にも、汎用計算機のOSと同様の機能を実装する実施例や、専用の計算機にファームウェアを付加して実現する実施例も考えられる。

【0024】クライアントプログラムの動作する計算機1では、アプリケーションプログラムであるクライアントプログラム5が動作している。図1の例では、このクライアントプログラム5からのリード/ライト呼び出しをサーバ計算機2に伝えるため、ライブラリ11と呼ぶ専用のプログラムをクライアントプログラム5にリンクしてある。このライブラリ11は、クライアントプログラム5からのリード/ライト呼び出しに従って、適切なリード/ライト要求を生成し、ファイルサーバ10に伝える。ライブラリ11とファイルサーバ10との間の要求/応答のやりとりは後述する転送方式に従うが、要求の送信はこのライブラリ11が制御する。なお、ライブラリ11の代わりに、それと同様の機能を汎用計算機のOSに実装する実施例も考えられる。

【0025】以下、幾つかの実施例について説明するが、その前に、それらの前提となる、ファイルサーバ10で複数の要求を多重処理するための基本的な考え方を説明する。

【0026】1つの要求に着目した場合、その典型的な処理シーケンスは図2(A)に示すように、幾つかの中断を挟んだ一連のアクションの処理から成り立っている。即ち、要求の種類に応じて該当する初期アクションが決められており、その後の処理の進行に従い幾つかの中間アクションを経て、最終アクションにおいてクライアントプログラム5の動作する計算機1に応答を返却することによって、1つの要求の処理が完結する。従って、中断点において他の要求のアクションを処理することによって、要求の多重処理が実現できる。

【0027】ファイルサーバ10では、上述のようなアクションをキュー(アクションキュー)によって管理し、図2(B)に示すようなメインループを実行する。ファイルサーバ10は要求を受け取るための口(受信ポート)を持っており、ここを読むことにより要求を受け取る(S1)。次に、受け取った要求に該当する初期アクションをアクションキューの末尾に登録する(S2)。次に、アクションキューの先頭からアクションを

1つ取り出す(S3)。そして、そのアクションを処理し(S4)、ステップS1に戻る。このようなメインループによって、ファイルサーバ10は複数の要求のアクションを交互に処理し、要求の多重処理を実現する。

【0028】また、要求処理の中断点の1つに物理I/Oの発行がある。このとき、I/Oを非同期で発行してからアクション処理を終了し、別の要求のアクションを処理することによって、多重処理を実現する。非同期I/Oの完了時には非同期I/O完了ハンドラ(図示せず)が動作するが、この非同期I/O完了ハンドラによって図2(C)に示す処理が実行される。

【0029】先ず、I/Oを発行したアクションの続きのアクションを決定する。この決定方法は以下のように幾つか考えられる。

1) ハンドラの引数にアクションを指定して直接決定する方法。

2) I/Oを発行したアクションの続きのアクションをキューに登録しておき、ハンドラがキューを検索することにより該当するアクションを決定する方法。

【0030】次に、決定したアクションをアクションキューの先頭に登録する。この登録されたアクションは、その後ファイルサーバ10のメインループで取り出され処理される。

【0031】なお、非同期I/O完了ハンドラは、ファイルサーバ10の動作中の任意のタイミングで割り込んで起動するが、アクションキューは、ファイルサーバ10と非同期I/O完了ハンドラとが同時に操作しないように排他する。

【0032】以下、計算機1上で動作するクライアントプログラム5からサーバ計算機2に接続されている二次記憶装置4上のデータをアクセスする方法について、幾つかの実施例を挙げて説明する。

【0033】(実施例1) 本実施例では、クライアントプログラム5からリード/ライト呼び出しがあった場合、ライブラリ11は図3に示す処理を実行する。

【0034】先ず、クライアントプログラム5からのリード/ライト呼び出しを解析し、或る一定サイズのデータへのリード/ライト要求に分割する(L11)。例えば、或る一定サイズを8KByteとする場合、クライアントプログラム5からのリード/ライト呼び出しが、例えば24KByteのデータのリードあるいはライトであったとすると、要求されたデータの先頭から8KByte分に対するリード/ライト要求と、次の8KByte分に対するリード/ライト要求と、最後の8KByte分に対するリード/ライト要求に分割する。

【0035】次に、分割して得られた全てのリード/ライト要求を全部一度に計算機間接続網3を介してサーバ計算機2のファイルサーバ10に送信する(L12)。そして、サーバ計算機2から返されてくる応答を待つ(L13)。応答を待つとは、クライアントプログラム

5の持つ応答受信用のポートを読む処理であり、応答が受信されていなければ受信されるまで待つことになる。リード/ライト要求を複数送信した場合、応答は複数返ってくることになる。

【0036】次に、サーバ計算機2からの応答を受信すると、今までに返却された応答の数を調べ、全応答が返却されたかどうかを確認する(L14)。応答の数がリード/ライト要求の分割数より少ない、すなわち、全ての応答が返却されていなければ、再びステップL13に戻って応答を待つ。全ての応答が返却されていれば、クライアントプログラム5の呼び出しに対してリターンする。

【0037】他方、ファイルサーバ10では、ライブラリ11からリード/ライト要求を受け取り、受け取ったリード/ライト要求1つに対し、要求された二次記憶装置4上のデータにアクセスをした後、要求元に応答1つを送信するという処理を、複数のリード/ライト要求に対し、多重化して実行する。要求を多重化して実行する方法は前述したので、ここでは1つのリード/ライト要求に着目したシーケンスを図4を参照して説明する。

【0038】ファイルサーバ10は、まず、当該リード/ライト要求にかかるリード/ライトデータを一旦格納するための中継となるメモリ(データバッファ)を確保する(S11)。次に、二次記憶装置4に物理リード/ライトを非同期で発行し、その完了を待つ(S12)。ここで最初のアクションが終了し、要求処理としては中断する。次に、物理リード/ライトが終了すると、リード/ライト応答を要求元に送信する(S13)。これがリード/ライト要求の最終アクションとなる。

【0039】このように本実施例では、クライアントプログラム5によるリード/ライト呼び出しを一定サイズのデータへのリード/ライト要求に分割して全部一度にサーバ計算機2に送信し、この送信されてきた複数のリード/ライト要求をサーバ計算機2が多重処理するようにしており、サーバ計算機2での二次記憶装置の物理リード/ライト処理と計算機間接続網3での通信処理が並行して行われることになる。このことによって、通信の時間が物理リード/ライト処理の時間と重なり、全体の処理時間が短縮される。

【0040】通信処理の時間と物理リード/ライト処理の時間とが重なる様子を図5を用いて説明する。図5では、クライアントプログラム5からのリード/ライト呼び出しを4つのリード/ライト要求に分割して要求した場合を示す。まず、ライブラリ11は最初に4つのリード/ライト要求をサーバ計算機2に一度に送る。すると、サーバ計算機2のファイルサーバ10では4つのリード/ライト要求を多重処理するが、最初の要求を受けた時点で物理リード/ライトが開始されるため、後続の要求の通信時間が最初のリード/ライト処理時間と重なる。これによって、要求の通信時間のほとんどがクライ

アントプログラム5から見えなくなる。さらに、各要求の終了時には応答をライブラリ11に返却することになるが、その間、サーバ計算機2のファイルサーバ10では、次の要求の物理リード/ライトを重ねて処理する。これによって、応答の通信時間のほとんどがクライアントプログラム5から見えなくなる。

【0041】図5を見ると、通信の時間のほとんどがサーバ計算機2での物理リード/ライトの時間と重なっていることが判る。これによって、ほとんどの通信時間はクライアントプログラム5から見えなくなり、リード/ライト呼び出しの処理時間が短縮される。但し、図5中の通信時間24はクライアントプログラム5から見えなくなるが、最初の要求と最後の応答の通信時間23はクライアントプログラム5に見えてしまう。しかし、アクセスするデータが大量で分割数が多くなる程、この時間23は物理リード/ライトの時間22の総和に比べ相対的に小さくなり、リード/ライト呼び出しの処理時間21は、物理リード/ライトの時間22の総和と同等になる。

【0042】また、サーバ計算機2からクライアントプログラム5の動く計算機1に応答が次々と戻ってくる。これに対し、ライブラリ11は全ての要求の処理が完了したことを把握することによって、クライアントプログラム5からのリード/ライト呼び出しの完了をみきわめることができる。

【0043】(実施例2)本実施例では、クライアントプログラム5からリード/ライト呼び出しがあった場合、ライブラリ11は図6に示す処理を実行する。

【0044】まず、実施例1と同様に、クライアントプログラム5からのリード/ライト呼び出しを解析し、或る一定サイズのデータへのリード/ライト要求に分割する(L21)。

【0045】次に、分割数が予め定められたスレッシュOLD値より大きいかどうかを調べる(L22)。そして、分割数がスレッシュOLD値より大きくない場合は、分割して得られた全てのリード/ライト要求を全部一度に計算機間接続網3を介してサーバ計算機2のファイルサーバ10に送信し(L24)、分割数がスレッシュOLD値より大きい場合は、分割して得られたリード/ライト要求のうち、スレッシュOLD値に等しい個数のリード/ライト要求だけを一度に計算機間接続網3を介してサーバ計算機2のファイルサーバ10に送信する(L23)。このとき、未だ送信されなかったリード/ライト要求は内部に保持しておく。そして、サーバ計算機2から返されてくる応答を待つ(L25)。

【0046】次に、サーバ計算機2からの応答を受信すると、未だ送信していないリード/ライト要求が残っているかどうかを確認する(L26)。未だ送信していないリード/ライト要求が残っている場合は、その内の1つのリード/ライト要求をサーバ計算機2のファイルサ

10

20

30

40

50

サーバ10に送信し、ステップL25の応答待ちに移る。ステップL26の確認で未送信リード/ライト要求が残っていないと判断した場合は、今までに返却された応答の数を調べ、全応答が返却されたかどうかを確認する(L28)。応答の数がリード/ライト要求の分割数より少ない、すなわち、全ての応答が返却されていなければ、再びステップL25に戻って応答を待つ。全ての応答が返却されていれば、クライアントプログラム5の呼び出しに対してリターンする。

【0047】ファイルサーバ10の処理は実施例1と同じであり、ライブラリ11からリード/ライト要求を受け取り、受け取ったリード/ライト要求1つに対し、要求された二次記憶装置4上のデータにアクセスした後、要求元に応答1つを送信するという処理を、複数のリード/ライト要求に対し、多重化して実行する。

【0048】本実施例の基本的な効果は、実施例1と同様に、通信時間のほとんどを物理リード/ライトと重ねてクライアントプログラムから見えなくするという効果を含んでいる。ただし、実施例1では分割して得られた全てのリード/ライト要求を全部一度に送信するため、クライアントプログラム5からのリード/ライト呼び出しのデータ量が大きい場合、一度に送信するリード/ライト要求の数が多くなり、サーバ計算機2および計算機間通信網3の負荷が過大になる。サーバ計算機2等の負荷が異常に高まると、リード/ライト要求の幾つかが消滅し処理されなくなる場合が生じるため、クライアントプログラム5からのリード/ライト呼び出しが失敗してしまうことがある。

【0049】これに対して本実施例では、サーバ計算機2で一度に受け取ることのできる要求数としてスレッシュOLD値を設定し、最初に送信するリード/ライト要求数をスレッシュOLD値以下に抑えているため、サーバ計算機2での最初の負荷を一定値以下に抑えることができる。さらに、1つの応答に対して1つのリード/ライト要求を送信するので、サーバ計算機2で受け取っているリード/ライト要求数は常にスレッシュOLD値以下となり、引続きサーバ計算機2での負荷を一定値以下に抑えることができる。

【0050】本実施例におけるサーバ計算機2とクライアントプログラム5との間で発生する要求/応答のやり取りの一例を図7に示す。この例では、クライアントプログラム5からのリード/ライト呼び出しが8つのリード/ライト要求に分割され、またスレッシュOLD値は4に設定されている。

【0051】まず、スレッシュOLD値が4なので、ライブラリ11は最初に4つだけのリード/ライト要求をサーバ計算機2に送信している。これにより、サーバ計算機2が最初に受け取るリード/ライト要求の数を4以下に抑えることができる。

【0052】その後、サーバ計算機2のファイルサーバ

10では個々のリード/ライト要求を多重処理し、1つずつ応答を返却する。これに対し、ライブラリ11では、1つの応答を受ける毎に、1つのリード/ライト要求をサーバ計算機2に送信する。これにより、サーバ計算機2で多重処理しているリード/ライト要求の数は常に4つ以下となる。

【0053】(実施例3) 本実施例では、クライアントプログラム5からリード/ライト呼び出しがあった場合、ライブラリ11は図8に示す処理を実行する。

【0054】まず、実施例1と同様に、クライアントプログラム5からのリード/ライト呼び出しを解析し、或る一定サイズのデータへのリード/ライト要求に分割する(L31)。

【0055】次に、この分割数を指定した予約要求を計算機間接続網3を介してサーバ計算機2のファイルサーバ10に送信し、その応答を待つ(L32)。

【0056】次に、サーバ計算機2のファイルサーバ10から上記の予約要求に対する応答を受信すると、その応答に含まれる予約数を取り出し、スレッシュOLD値とする(L33)。

【0057】次に、ステップL31での分割数がこのスレッシュOLD値より大きいかどうかを調べる(L34)。そして、分割数がスレッシュOLD値より大きくない場合は、分割して得られた全てのリード/ライト要求を全部一度に計算機間接続網3を介してサーバ計算機2のファイルサーバ10に送信し(L36)、分割数がスレッシュOLD値より大きい場合は、分割して得られたリード/ライト要求のうち、スレッシュOLD値に等しい個数のリード/ライト要求だけを一度に計算機間接続網3を介してサーバ計算機2のファイルサーバ10に送信する(L35)。このとき、未だ送信されなかったリード/ライト要求は内部に保持しておく。そして、サーバ計算機2から返されてくる応答を待つ(L37)。

【0058】次に、サーバ計算機2からの応答を受信すると、未だ送信していないリード/ライト要求が残っているかどうかを確認する(L38)。未だ送信していないリード/ライト要求が残っている場合は、その内の1つのリード/ライト要求をサーバ計算機2のファイルサーバ10に送信し(L39)、ステップL37の応答待ちに移る。ステップL38の確認で未送信リード/ライト要求が残っていないと判断した場合は、今までに返却された応答の数を調べ、全応答が返却されたかどうかを確認する(L3a)。

応答の数がリード/ライト要求の分割数より少ない、すなわち、全ての応答が返却されていなければ、再びステップL37に戻って応答を待つ。

【0059】全ての応答が返却されていれば、サーバ計算機2における予約を解除するための要求(予約解除要求)を計算機間接続網3を介してサーバ計算機2のファイルサーバ10に送信し、その応答を待つ(L3b)。

そして、予約解除要求に対する応答が返却された後、ク

10

20

30

40

50

クライアントプログラム5の呼び出しに対してリターンする。

【0060】本実施例の場合、ファイルサーバ10は、リード/ライト要求処理以外に、予約要求処理と予約解除要求処理とを行う。リード/ライト要求処理は実施例1、2とほぼ同じなので、以下、予約要求処理と予約解除要求処理とを説明する。

【0061】図9は1つの予約要求処理に着目したシーケンスを示している。前述したようにライブラリ11から予約要求が送信されてくると、ファイルサーバ10は、先ず、当該クライアントプログラムに対して許容できるリード/ライト要求の数を計算する(S31)。

【0062】この許容数は現在のファイルサーバ10の負荷状況によって決まる。具体的には、ファイルサーバ10がリード/ライト要求を処理するために使える現時点のメモリ残量によってほぼ決まる。例えば、ファイルサーバ10がリード/ライト要求を処理するために使える全メモリ量をAとし、1つのリード/ライト要求を処理するために必要なメモリ量をBとすると、 $[A/B]$ がファイルサーバ10の処理負荷上限となる。また、現時点でC個のリード/ライト要求の予約を受け付けている場合、現在の許容数の総計(総許容数)はCなので、現時点でのリード/ライト要求の許容数は、処理負荷上限-Cである。そこで、ファイルサーバ10では、処理負荷の上限となる数から、現在の許容数の総計(総許容数)を引いたものを、当該クライアントプログラムに対する許容数として求める。

【0063】次に、ステップS31で求めた許容数と予約要求で指定された分割数のうち小さい方の値を予約数として決定する(S32)。そして、現在の許容数の総計を今回の予約数の分だけ増やし(S33)、ステップS32で求めた予約数を指定した応答を要求元に送信する(S34)。今回の予約数の分だけ現在の許容数の総計を増やすことにより、今回の予約数に見合ったメモリ量を他のクライアントプログラムから発生したリード/ライト要求処理で使われないようにすることができる。従って、当該クライアントプログラムからのリード/ライト要求にかかるリード/ライトデータを一旦格納するための中継となるメモリを確保する際にメモリ不足が生じるのを防止できる。

【0064】なお、ステップS32の処理内容を、予約数 $=\min$ (許容数、分割数、1クライアントプログラム当たりに付与する最大許容数)に変更する実施例も考えられる。こうすれば、1クライアントプログラムに付与する最大許容数を制限でき、1つのクライアントプログラムが現時点の許容数を全て占有してしまうことを防止できる。

【0065】以上のような予約要求処理は1つのアクションから成り立つ。

【0066】他方、図10は1つの予約解除要求の処理

に着目したシーケンスを示している。前述したようにライブラリ11から予約解除要求が送信されてくると、ファイルサーバ10は現在の許容数の総計から、当該クライアントプログラムに与えていた予約数を差し引いて更新する(S35)。これにより予約されていたメモリ量が解放され、他のクライアントプログラムからのリード/ライト要求の処理のために使用できるメモリ量がその分だけ増加する。そして、予約解除要求処理の終了を応答によってライブラリ11に伝える(S36)。このような予約解除要求処理は1つのアクションから成り立つ。

【0067】実施例2のように、サーバ計算機2の処理能力によってスレッシュホールド値を予め決めておいた場合、クライアントプログラムが1つならば、サーバ計算機2の処理能力を超えた数のリード/ライト要求が来ることはない。しかし、サーバ計算機2に対してリード/ライト要求を送信するクライアントプログラムが複数存在する場合、サーバ計算機2の処理能力を超えてしまうことがある。

【0068】これに対して本実施例では、動的に変化するサーバ計算機2のファイルサーバ10の負荷状況に応じて、1つのクライアントプログラムに対して許容する予約数をライブラリ11に知らせ、ライブラリ11はその予約数をスレッシュホールド値として、以降、このスレッシュホールド値に従ってリード/ライト要求を行うため、サーバ計算機2に発生する負荷が一定値を超えないように制御することができる。つまり、サーバ計算機2が複数のクライアントプログラムを相手にしても、サーバ計算機2の処理能力を超えることはない。また、クライアントプログラムからのリード/ライト要求が全て終了した際には予約数がリセットされるため、以降に発生するクライアントプログラムの予約要求に対してメモリを有効に活用することができる。

【0069】本実施例におけるサーバ計算機2とクライアントプログラム5との間で発生する要求/応答のやり取りの一例を図11に示す。予約処理(イ)と予約解除処理(ロ)以外は実施例2と同様のやり取りとなる。図11の例では、クライアントプログラム5からのリード/ライト呼び出しを5つのリード/ライト要求に分割しているが、ファイルサーバ10からの予約数が3であった場合を想定している。このため、ライブラリ11はスレッシュホールド値に3を設定し、最初に3つのリード/ライト要求をサーバ計算機2に送っている。以後の処理は実施例2の処理手順と同様であるが、全てのリード/ライト要求を処理した後、予約解除処理(ロ)が行われ、クライアントプログラム5からのリード/ライト呼び出しへ戻っている。

【0070】(実施例4)本実施例では、クライアントプログラム5からリード/ライト呼び出しがあった場合、ライブラリ11は図12に示す処理を実行する。

10

20

30

40

50

【0071】先ず、実施例1と同様に、クライアントプログラム5からのリード/ライト呼び出しを解析し、或る一定サイズのデータへのリード/ライト要求に分割する(L41)。

【0072】次に、この分割数が1より大きいかどうかを調べる(L42)。分割数が1つなら、予約要求することなく、その1つのリード/ライト要求を計算機間接続網3を介してサーバ計算機2のファイルサーバ10に送信し、応答の受信を待つ(L44)。

【0073】他方、上記の分割数が1より大きいときは、分割して得られたリード/ライト要求のうちの1つのリード/ライト要求に予約要求を付加した予約+リード/ライト一括要求を、1つの要求としてサーバ計算機2のファイルサーバ10に送信し、予約応答の受信を待つ(L43)。予約応答の受信を待つとは、クライアントプログラム5の持つ応答受信用のポートを読む処理であり、予約応答が受信されていなければ受信されるまで待つことになる。

【0074】次に、サーバ計算機2から予約の応答が返却されると、その応答に含まれる予約数を取り出し、スレッシュOLD値とする(L45)。

【0075】次に、ステップL41での分割数-1がこのスレッシュOLD値より大きいかどうかを調べる(L46)。そして、分割数-1がスレッシュOLD値より大きくない場合は、分割して得られた残りの全てのリード/ライト要求を全部一度に計算機間接続網3を介してサーバ計算機2のファイルサーバ10に送信する(L48)。但し、その内の1つのリード/ライト要求は、予約解除要求を付加して、予約解除+リード/ライト一括要求とし、これを1つの要求として送信する。また、分割数-1がスレッシュOLD値より大きい場合は、分割して得られた残りのリード/ライト要求のうち、スレッシュOLD値に等しい個数のリード/ライト要求だけを一度に計算機間接続網3を介してサーバ計算機2のファイルサーバ10に送信する(L47)。このとき、未だ送信されなかったリード/ライト要求は内部に保持しておく。そして、サーバ計算機2から返されてくるリード/ライトの応答を待つ(L49)。

【0076】次に、サーバ計算機2からのリード/ライトの応答を受信すると、未だ送信していないリード/ライト要求が残っているかどうかを確認する(L4a)。未だ送信していないリード/ライト要求が残っている場合は、その数が1つかどうかを調べ(L4c)、1つでなければ、即ち未だ送信していないリード/ライト要求が2つ以上残っている場合は、その内の1つのリード/ライト要求をサーバ計算機2のファイルサーバ10に送信し(L4d)、ステップL49の応答待ちに移る。他方、未だ送信していないリード/ライト要求が1つの場合は、それに予約解除要求を付加して予約解除+リード/ライト一括要求とした1つの要求をサーバ計算機2の

ファイルサーバ10に送信し(L4e)、ステップL49の応答待ちに移る。

【0077】そして、ステップL4aの確認で未送信リード/ライト要求が残っていないと判断した場合は、今までに返却された応答の数を調べ、全応答が返却されたかどうかを確認し(L4b)、応答の数がリード/ライト要求の分割数より少ない、すなわち、全ての応答が返却されていなければ、再びステップL49に戻って応答を待つ。全ての応答が返却されていれば、クライアントプログラム5の呼び出しに対してリターンする。

【0078】本実施例の場合、ファイルサーバ10は、リード/ライト要求処理以外に、予約+リード/ライト一括要求処理と予約解除+リード/ライト一括要求処理とを、多重化して実行する。

【0079】図13は1つのリード/ライト要求に着目したシーケンスを示している。ファイルサーバ10は、先ず、当該リード/ライト要求にかかるリード/ライトデータを一旦格納するための中継となるメモリを確保する(S41)。次に、二次記憶装置4に非同期で物理リード/ライトを行い、その完了を待つ(S42)。

【0080】次に、物理リード/ライトが終了すると、同じクライアントプログラム5からの要求で、予約解除+リード/ライト一括要求の処理が中断しているかどうかを調べる(S43)。若し、中断しているならば、同じクライアントプログラム5からの処理中のリード/ライト要求が他にあるかどうかを調べ(S44)、あれば、中断している予約解除+リード/ライト一括要求を要求キューに登録し(S45)、前記物理リード/ライトの終了したリード/ライト要求に対する応答を要求元に送信する(S46)。他方、予約解除+リード/ライト一括要求の処理が中断していないか或いはそのような予約解除+リード/ライト一括要求が存在しない場合、あるいは中断している予約解除+リード/ライト一括要求があっても同じクライアントプログラムからの処理中のリード/ライト一括要求があれば、ステップS45をスキップして、当該リード/ライト要求の応答を要求元に送信する(S46)。

【0081】図14は1つの予約+リード/ライト一括要求の処理に着目したシーケンスを示している。前述したようにライブラリ11から予約+リード/ライト一括要求が送信されてくると、ファイルサーバ10は、先ず、当該クライアントプログラムに対して許容できるリード/ライト要求の数を計算する(S47)。この許容数は現在のファイルサーバ10の負荷状況によって決まる。具体的には、ファイルサーバ10の処理負荷の上限となる数から、現在の許容数の総計(総許容数)を引いたものとして計算される。

【0082】次に、ステップS47で求めた許容数と予約要求で指定された分割数-1のうち小さい方の値を予約数として決定する(S48)。そして、現在の許容数

の総計を今回の予約数の分だけ増やしておき（S49）、ステップS48で求めた予約数を指定した応答を要求元に非同期で送信する（S4a）。

【0083】なお、ステップS48の処理内容を、予約数 = min（許容数、分割数 - 1、1クライアントプログラム当たり付与する最大許容数）に変更する実施例も考えられる。こうすれば、1クライアントプログラムに付与する最大許容数を制限でき、1つのクライアントプログラムが現時点の許容数を全て占有してしまうことを防止できる。

【0084】ここまでの処理は実施例3のファイルサーバ10での予約要求処理とはほぼ同様である。但し、本実施例では、予約応答を非同期で送信した後、引き続きリード／ライト要求の処理に移るため、リード／ライト要求の処理の時間と予約応答の通信時間とが重なる。

【0085】リード／ライト要求の処理では、まず、当該リード／ライト要求にかかるリード／ライトデータを一旦格納するための中継となるメモリを確保する（S4b）。次に、二次記憶装置4に非同期で物理リード／ライトを行い、その完了を待つ（S4c）。そして、物理

リード／ライトが終了すると、リード／ライト応答を要求元に送信する（S4d）。
 【0086】他方、図15は1つの予約解除＋リード／ライト一括要求に着目したシーケンスを示している。前述したようにライブラリ11から予約解除＋リード／ライト一括要求が送信されてくると、先ずリード／ライト要求処理を行い、次いで予約解除処理を行う。リード／ライト要求処理では、当該リード／ライト要求にかかるリード／ライトデータを一旦格納するための中継となるメモリを確保し（S4e）、二次記憶装置4に非同期で物理リード／ライトを行い、その完了を待つ（S4f）。物理リード／ライトが終了すると、同じクライアントプログラム5からの処理中のリード／ライト要求が他にあるかどうかを調べ（S4g）、若しあれば、処理中のリード／ライト要求がなくなるまで中断する。中断の方法は、当該要求を中断キューに登録することで行う。中断キューに登録された要求は、前述の図13のステップS45で中断キューから要求キューに登録しなおされる。

【0087】他に処理中のリード／ライト要求がなくなると、予約解除の処理となり、現在の総許容数から予約数を差し引いて更新する（S4h）。これにより当該クライアントプログラムに対して予約されていたメモリ量が解放される。そして、リード／ライト応答に予約解除の応答を含めて要求元に送信する（S4i）。

【0088】本実施例によれば、予約＋リード／ライト一括要求を導入したので、ライブラリ11側では予約要求を別要求として最初に送らずに済むとともに、ファイルサーバ10側では予約応答に引き続いてリード／ライト処理が行われるので、予約応答の通信時間を物理リー

ド／ライトの時間と重ねることができる。また、サーバ計算機での物理リード／ライトの最中に、クライアントプログラムから後続のリード／ライト要求の送信が行われれば、それらの通信時間も物理リード／ライトの時間に重ねることができる。

【0089】また、予約解除＋リード／ライト一括要求を導入したので、予約解除の要求を別要求として最後に送らずに済み、その分の通信時間を削減することができる。

10 【0090】以上のことから、最初と最後の予約／予約解除のための通信時間を削減することができる。図16はサーバ計算機2とクライアントプログラム5との間で発生する要求／応答のやり取りの一例を示している。最初の予約＋リード／ライト一括要求（ハ）は1要求としてサーバ計算機2に送られる。サーバ計算機2では、まず予約処理を行い、予約応答を返却し、同時にリード／ライト処理に移る。この際、物理リード／ライトと予約応答の通信時間が重なっている。

【0091】クライアントプログラム5の動作する計算機1のライブラリ11で予約応答を受け取った後のリード／ライト要求処理は、実施例3と同様である。ただし、最後の要求が予約解除＋リード／ライト一括要求となる点が異なる。最後の要求の処理は、図16に示す予約解除＋リード／ライト（ニ）の処理に相当する。図16の例では、クライアントプログラム5からのリード／ライト呼び出しを5つのリード／ライト要求に分割しているが、ファイルサーバ10からの予約数が3であった場合を想定している。このため、ライブラリ11はスレッシュホールド値に3を設定し、予約応答受信後、最初に3つのリード／ライト要求をサーバ計算機2に送っている。以後の処理は実施例2、3の処理手順と同様であるが、リード／ライト要求の残りが1つになったとき、予約解除とリード／ライトを一緒にした要求（予約解除＋リード／ライト一括要求）を行う。サーバ計算機では、この要求に対してリード／ライトの処理と予約解除の処理を行い、1つの応答として返却する。これにより、最後の通信時間を削減することができる。

【0092】以上本発明の実施例について説明したが、本発明は以上の実施例にのみ限定されずその他各種の付加変更が可能である。例えば、図1では、1つのサーバ計算機2に、1つのクライアントプログラム5が動作する計算機1を計算機間通信網3を介して接続しているが、1つ又は複数のクライアントプログラム5が動作する計算機を複数台、計算機間通信網3に接続した計算機システムに対しても適用可能である。また、サーバ計算機2には1つの二次記憶装置4のみ接続されている例を示したが、複数の二次記憶装置が接続されていても良い。

【0093】

50 【発明の効果】以上説明したように本発明によれば以下

のような効果を得ることができる。

【0094】請求項1ないし4記載の発明によれば、遠隔二次記憶装置へのデータアクセス時における通信時間のほとんどをクライアントプログラムから見えなくし、あたかもその二次記憶装置がクライアントプログラムの動作する計算機に接続されているかのようなアクセス時間を達成することができる。

【0095】また請求項2記載の発明によれば、クライアントプログラムが大容量のデータをアクセスする場合であっても、即ち分割されたリード/ライト要求の数が多くなる場合であっても、一度に要求するリード/ライトの量を抑制でき、第2の計算機での処理負荷が過大にならないようにすることができる。

【0096】更に請求項3記載の発明によれば、1台の第2の計算機に対して複数のクライアントプログラムがリード/ライト要求を行う場合でも、第2の計算機の負荷に応じて一度に要求するリード/ライトの量を抑制でき、第2の計算機の処理負荷が過大にならないようにすることができる。

【0097】また更に請求項4記載の発明によれば、第1の計算機側では予約要求を別要求として最初に送らずに済み、第2の計算機側では予約応答に引き続いてリード/ライト処理が行われるので、予約応答の通信時間を物理リード/ライトの時間と重ねることができる。また、第2の計算機での物理リード/ライトの最中に、第1の計算機から後続のリード/ライト要求の送信が行われれば、それらの通信時間も物理リード/ライトの時間に重ねることができる。さらに、第1の計算機側では予約解除の要求を別要求として最後に送らずに済み、その分の通信時間を削減することができる。

【図面の簡単な説明】

【図1】本発明を適用した計算機システムの一例を示すブロック図である。

【図2】ファイルサーバで複数の要求を多重処理するための処理の一例を示すフローチャートである。

【図3】実施例1におけるライブラリの処理の一例を示すフローチャートである。

【図4】実施例1におけるファイルサーバの、1つのリード/ライト要求に着目した処理の一例を示すフローチャートである。

【図5】実施例1におけるクライアントプログラムとサーバ計算機との要求/応答のやり取りを示すタイミング

チャートである。

【図6】実施例2におけるライブラリの処理の一例を示すフローチャートである。

【図7】実施例2におけるクライアントプログラムとサーバ計算機との要求/応答のやり取りを示すタイミングチャートである。

【図8】実施例3におけるライブラリの処理の一例を示すフローチャートである。

【図9】実施例3におけるファイルサーバの、1つの予約要求に着目した処理の一例を示すフローチャートである。

【図10】実施例3におけるファイルサーバの、1つの予約解除要求に着目した処理の一例を示すフローチャートである。

【図11】実施例3におけるクライアントプログラムとサーバ計算機との要求/応答のやり取りを示すタイミングチャートである。

【図12】実施例4におけるライブラリの処理の一例を示すフローチャートである。

【図13】実施例4におけるファイルサーバの、1つのリード/ライト要求に着目した処理の一例を示すフローチャートである。

【図14】実施例4におけるファイルサーバの、1つの予約+リード/ライト一括要求に着目した処理の一例を示すフローチャートである。

【図15】実施例4におけるファイルサーバの、1つの予約解除+リード/ライト一括要求に着目した処理の一例を示すフローチャートである。

【図16】実施例4におけるクライアントプログラムとサーバ計算機との要求/応答のやり取りを示すタイミングチャートである。

【図17】遠隔二次記憶装置への従来のアクセス方法の説明図である。

【符号の説明】

1…クライアントプログラムの動作する計算機

2…サーバ計算機

3…計算機間接続網

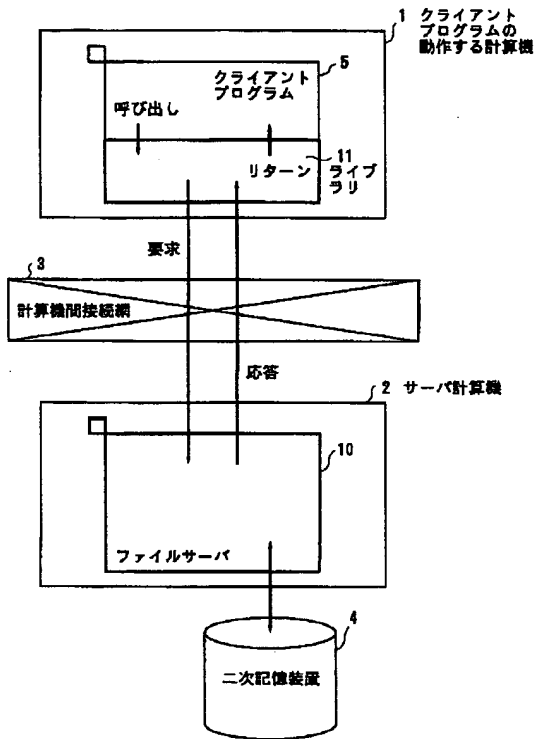
4…二次記憶装置

5…クライアントプログラム

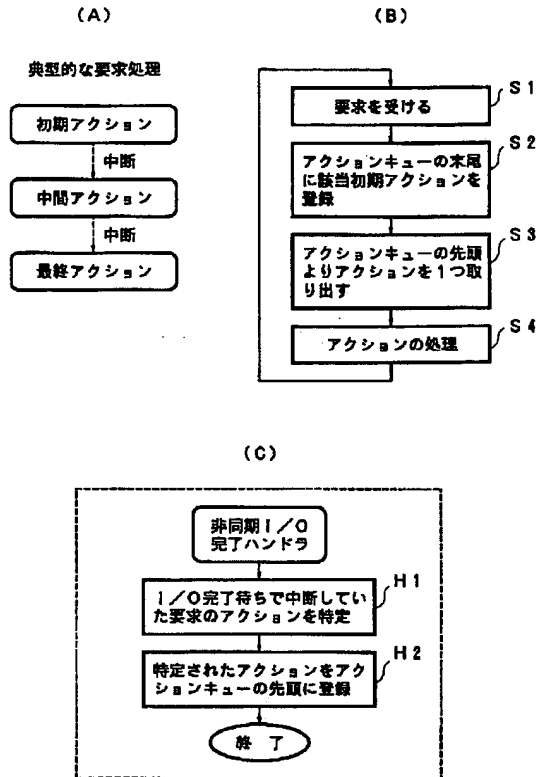
10…ファイルサーバ

11…ライブラリ

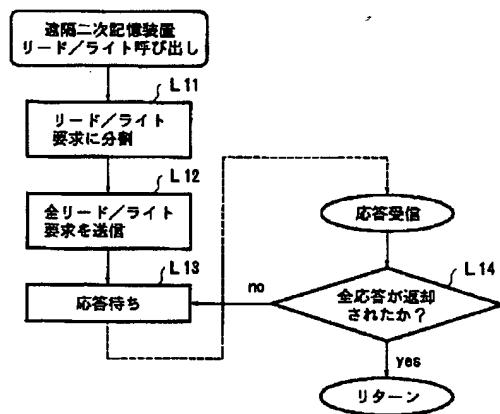
【図1】



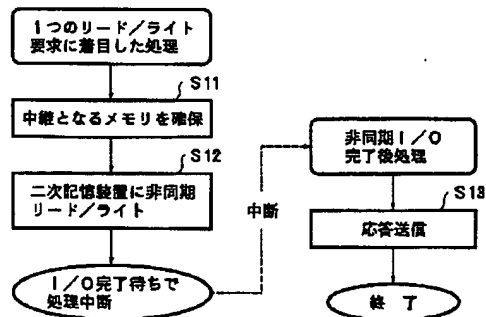
【図2】



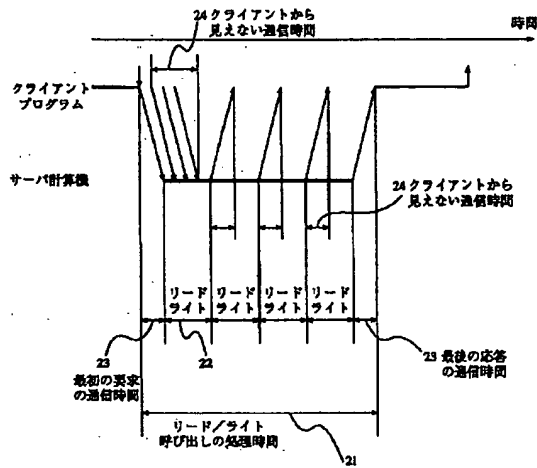
【図3】



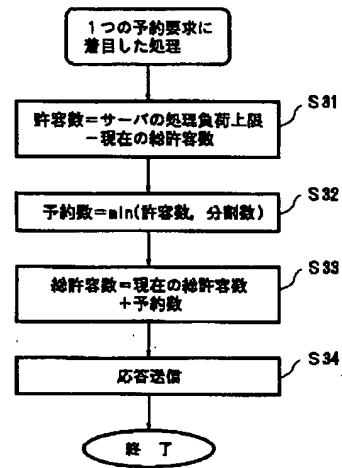
【図4】



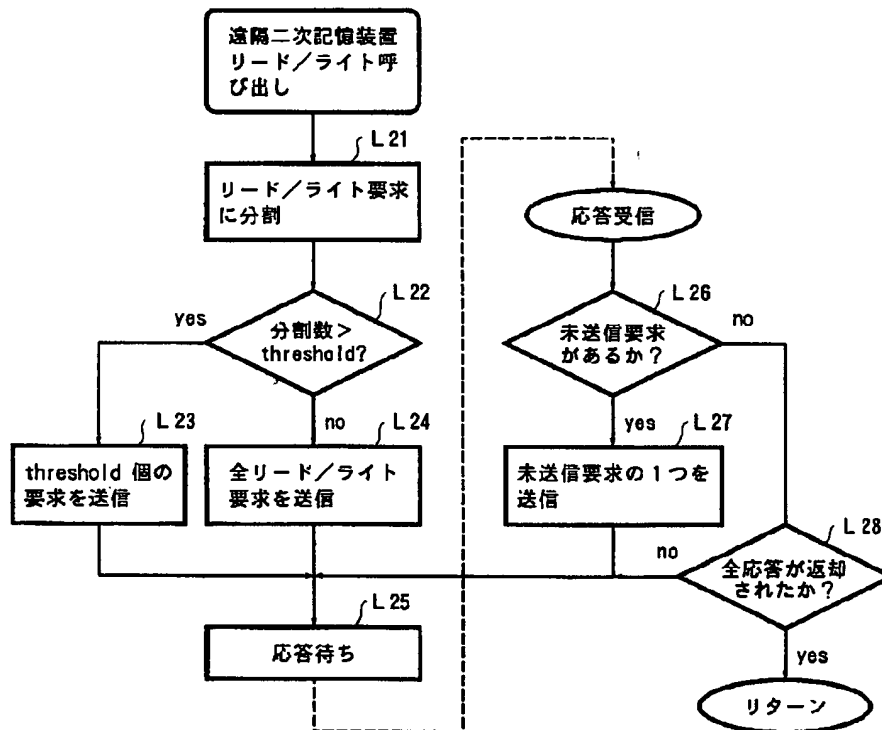
【図5】



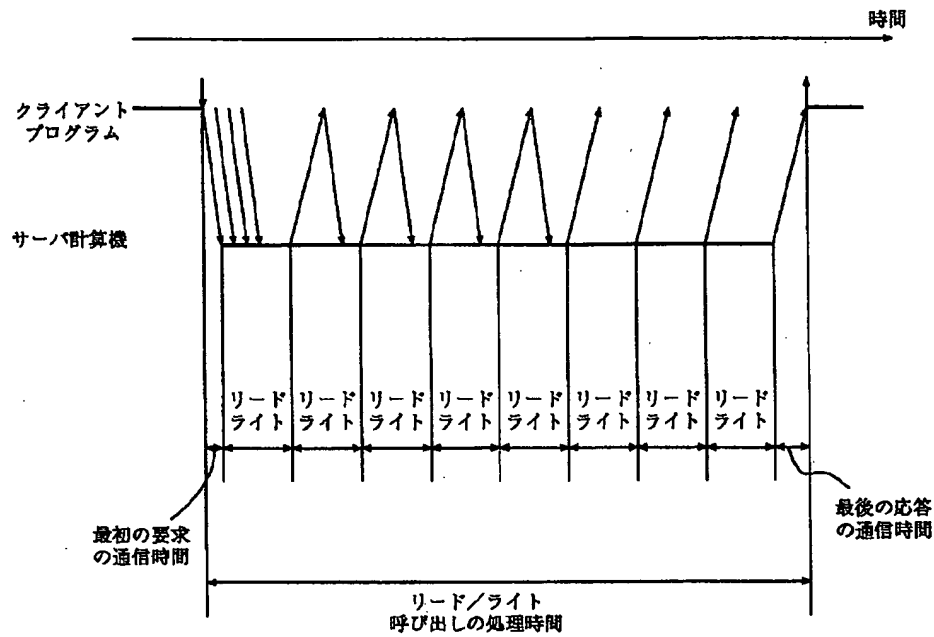
【図9】



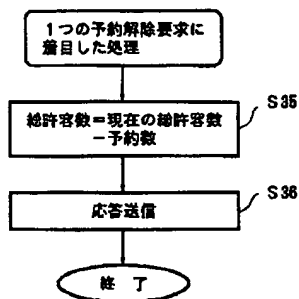
【図6】



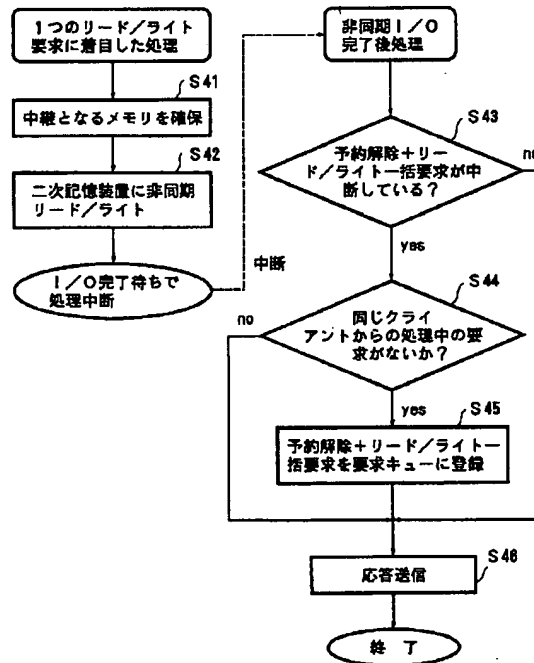
【図7】



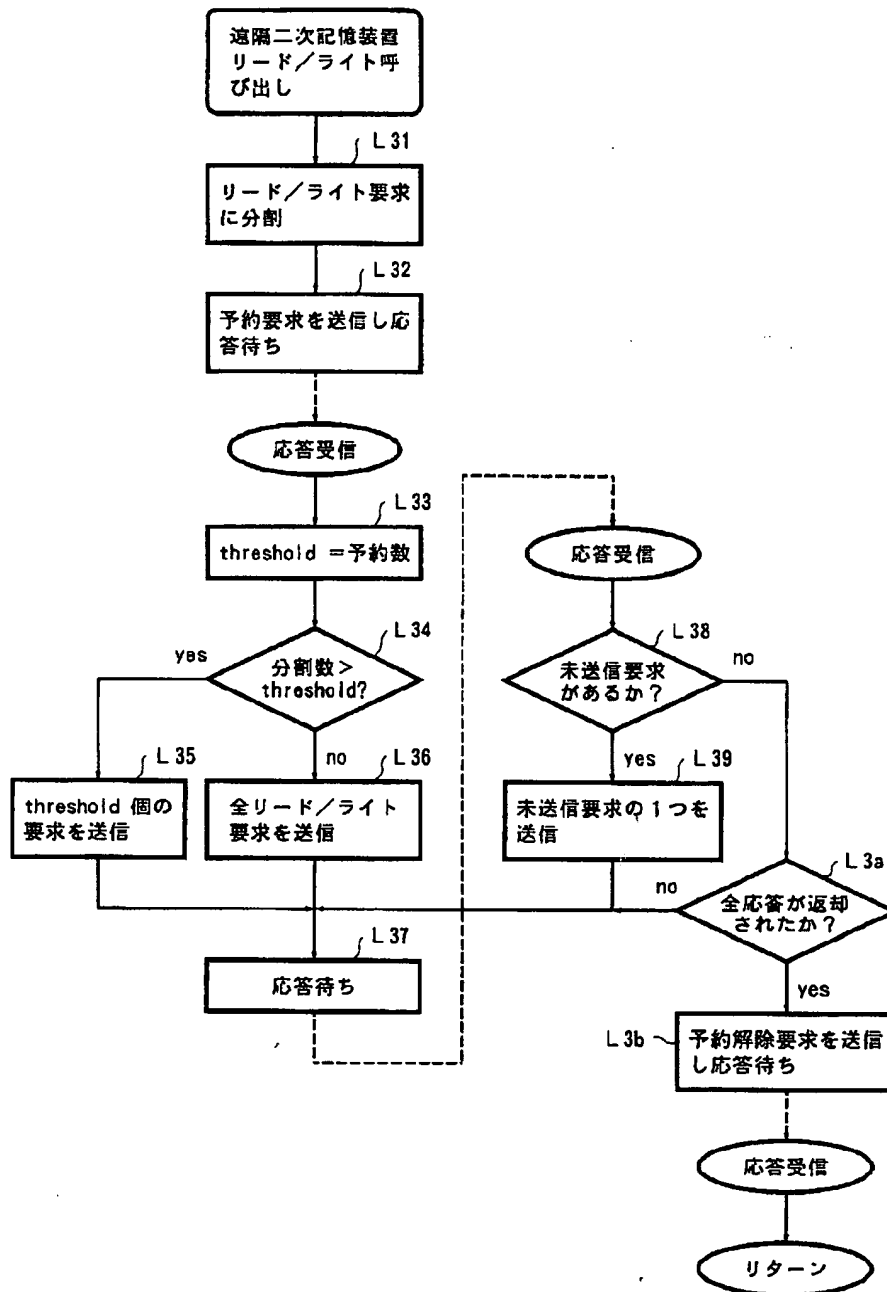
【図10】



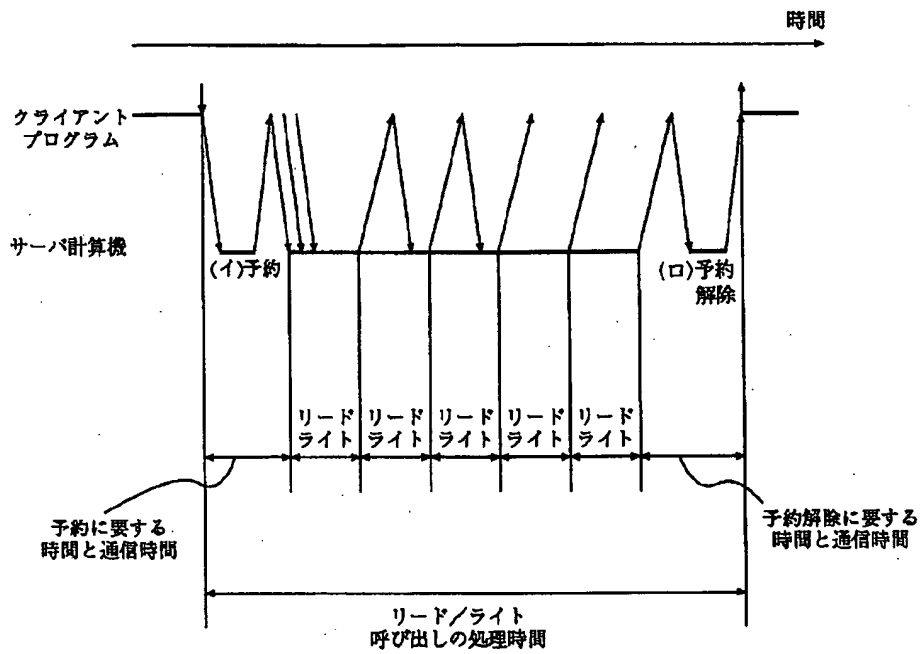
【図13】



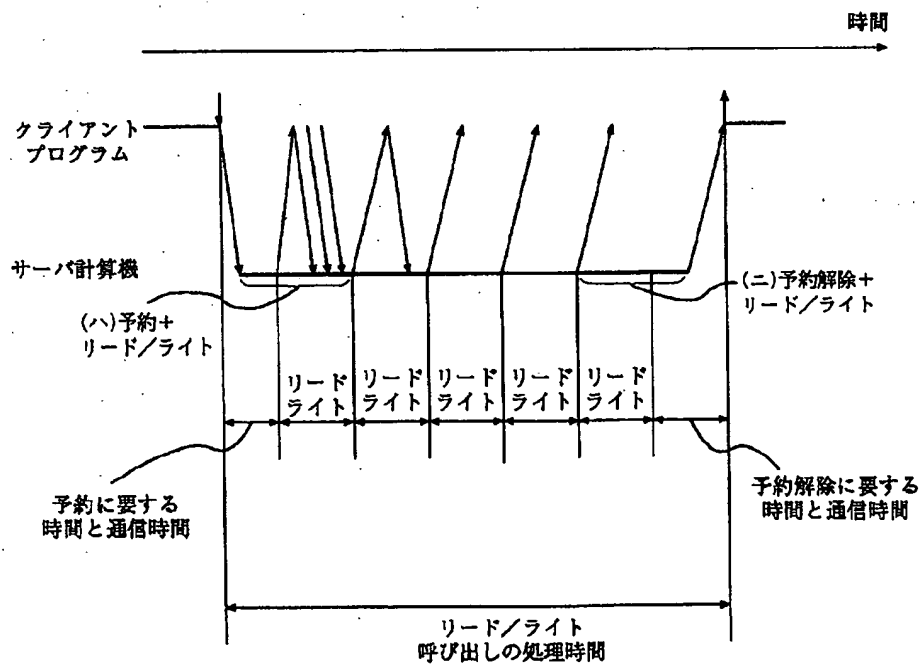
【図8】



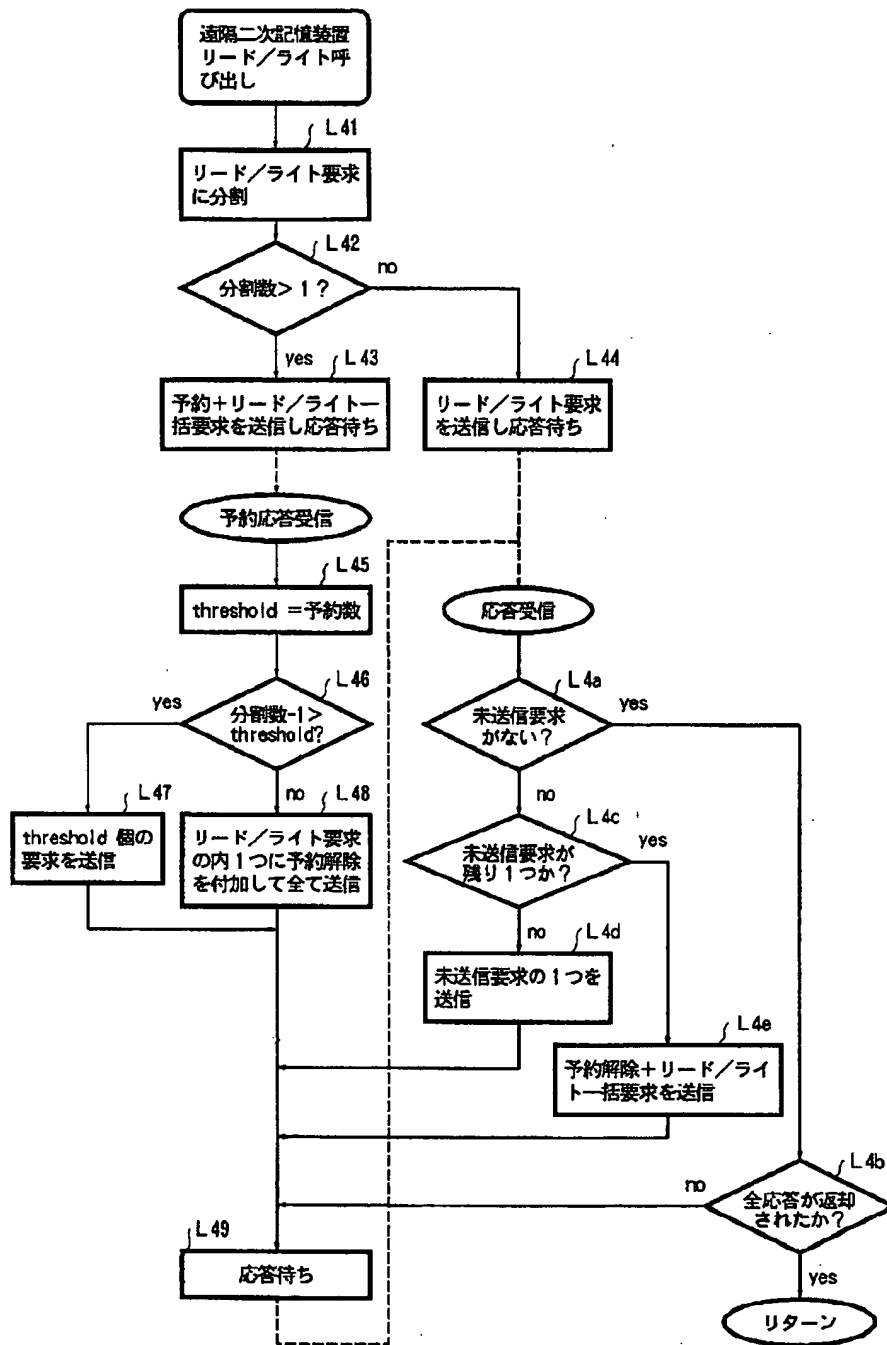
【図11】



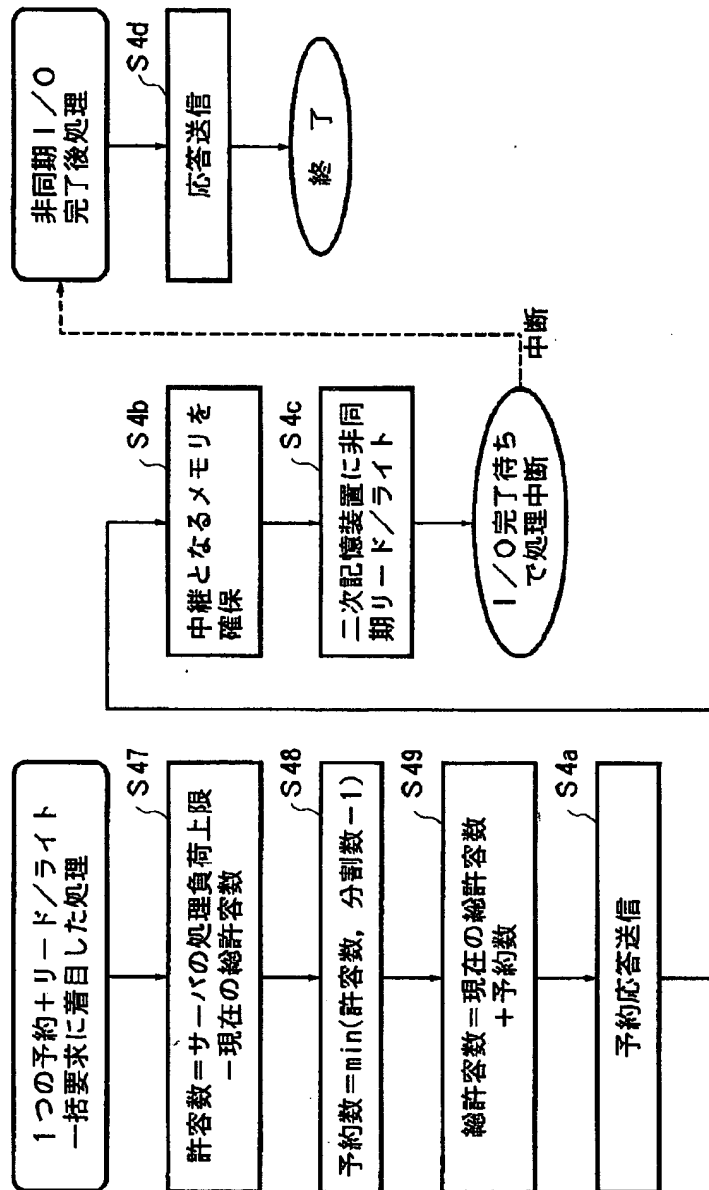
【図16】



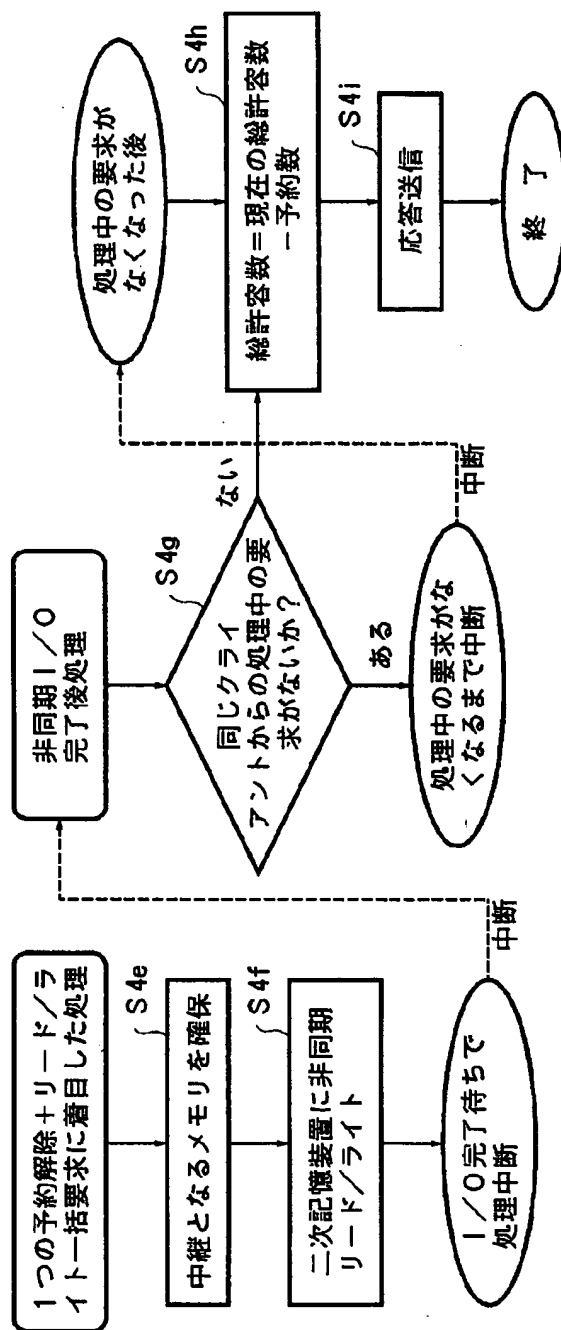
【図12】



【図14】

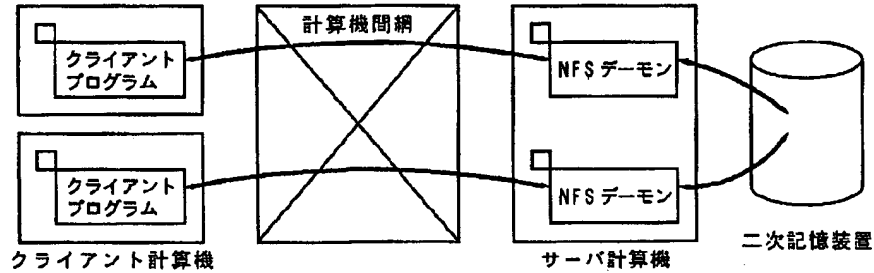


【圖 15】

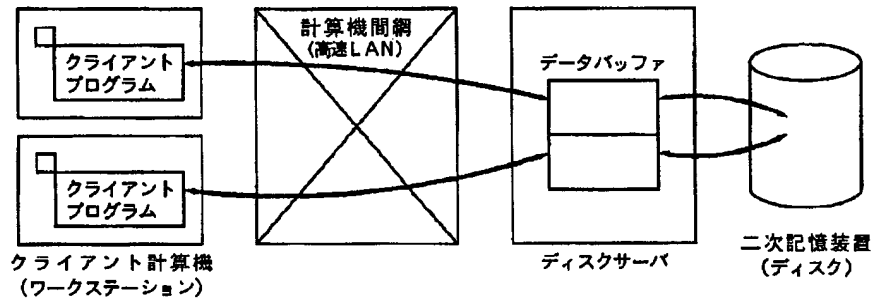


【図17】

(A)



(B)



(C)

